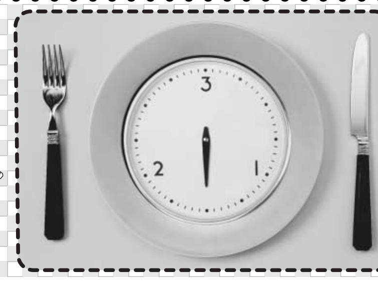
Cベース設計

3分間クッキング



第9回　forkの使いかた

鮫島正裕

ロンドンでイタリア料理店に行ったときのことである．頼んだスパゲティが出てきて，さあ食べようと皿の左右を見てみると，ナイフとフォークが置いてあるのに気が付いた．そう，ここはイギリスなのでした．

イギリスのテーブル・マナーでは，フォークは左手，ナイフは右手と決まっていて，右手にフォークを持ちかえてはいけないらしい．いっしょにいた友人（右利き）もフォークを左手に持ってスパゲティを見つめて固まっている．うーんと思い，首をすくめて周りを見まわし，「スパゲティはイタリアのものだから，イタリア方式で…」ということで，そのときは右手にフォークを持って食べてしまった．

この後，気になってテーブル・マナーについて調べてみたのだが，イギリス方式ではフォークの先を上に向けることも禁止のようなので，これに従うと巻きつける動作ができない．必然的にフォークでスパゲティの山をさして，ナイフでスパゲティを切りつつ持ち上げて，口もとに運ぶことになる．残り少なくなってきたら，フォークの背にスパゲティをのせる．これでは，なんともスパゲティを食べるには効率が悪い．フランス方式では右手に持ち替えてもよく，フォークですくう使いかたもOKのようなので，スパゲティはずっと食べやすいと思う．

イギリスとフランスは，テーブル・マナーが制定された中世に反目しあっていたためなのか，スープでのすくいかたなど，違うことが多い．ただ，テーブル・マナーの違いの最大の理由は，その時代にマナーを定めた人々が食べていた物に個々に最適化したからなのだと思う．

＊　　　　＊　　　　＊

リスト1はUNIXが普及し始めたころ，その特徴を示すコードとしてよく見かけたもので，パイプラインを使って親プロセスと子プロセスの間でコミュニケーションをとるサンプルである．7～10行目で親子間の送受信用の`pipe`を作成し，11行目の`fork`で二手に分かれる．15～26行目の子ども側では，16，17行目で子どもの標準入力を`oya_ko`パイプに，18，19行目で子どもの標準出力を`ko_oya`パイプに`dup`でつなぎ変え，20～23行目で必要なくなったファイル・ディスクリプタを`close`する．そして，24行目で，逆ポーランド式の計算コマンドの`dc`を起動する．

リスト1　forkとpipeの使用例

```
 1  #include<stdio.h>
 2  #define ERR(x) fprintf(stderr,#x " error¥n")
 3  main()
 4  {
 5    char buf[16];
 6    int i;
 7    int oya_ko[2];
 8    int ko_oya[2];
 9    if( pipe(oya_ko) == -1) ERR(pipe);
10    if( pipe(ko_oya) == -1) ERR(pipe);
11    switch( fork()){
12      case -1:
13        ERR(cannot create new process);
14        break;
15      case  0: //ko
16        if( close(0) == -1 ) ERR(close ko in);
17        if( dup(oya_ko[0]) != 0 ) ERR(oya_ko dup);
18        if( close(1) == -1 ) ERR(close ko out);
19        if( dup(ko_oya[1]) != 1 ) ERR(ko_oya dup);
20        if( close(oya_ko[0]) == -1 ) ERR(close);
21        if( close(oya_ko[1]) == -1 ) ERR(close);
22        if( close(ko_oya[0]) == -1 ) ERR(close);
23        if( close(ko_oya[1]) == -1 ) ERR(close);
24        execlp("dc","dc", NULL);
25        ERR(exec);
26        exit(1);
27    }
28    if( close(oya_ko[0]) == -1 ) ERR(close);
29    if( close(ko_oya[1]) == -1 ) ERR(close);
30    write( oya_ko[1],"1¥n2¥n+¥np¥nq¥n",10);
31    i = read( ko_oya[0],buf,16);
32    buf[i] = '¥0';
33    printf( "%s",buf);
34    close( ko_oya[0]);
35    close( oya_ko[1]);
36  }
```

28～35行目は親側で，28，29行目で`pipe`の子ども側の入出力を`close`し，30行目で子ども（`dc`コマンド）に対して「1＋2を計算して出力し，終了せよ」と送信する．31行目で子どもからの返信を受け取り，33行目で答えの3を出力する．

＊　　　　＊　　　　＊

C言語にもデザイン・パターンやシステム記述のIEEE標準化といった作法（テーブル・マナー）が定められているが，相手（食べ物）がハードになったりソフトになったりする場合はどう対処すればよいのだろうか．

＜筆者プロフィール＞
鮫島正裕（さめしま・まさひろ）．高速ディジタル回路のコンサルティングやEDAツール関係の仕事に従事．